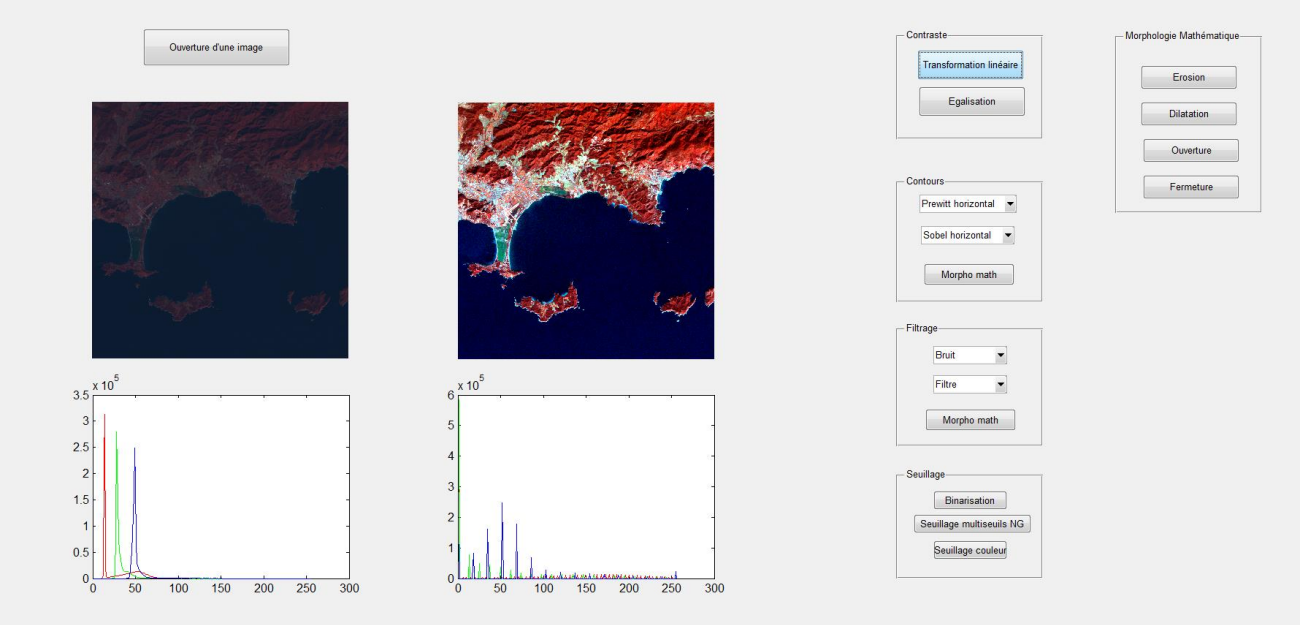
TP Instrumentation
Capteur Vision

SeaTech

IRIS 2ème Année



Audrey Minghelli
Durée des TP 3h

Généralités

Durant les 5 séances de TP sur Matlab, on vous demande de créer une interface homme machine (IHM) avec différentes fonctionnalités vues en cours.

Le compte rendu des TP est à déposer 2 semaines après la fin des TP sur le site Moodle de l'université de Toulon.

Initiation à Matlab en traitement des images

Exo. 1 : Généralités

1) L'image N/B est une variable à ….. dimensions
2) L'image couleur est une variable à ….. dimensions
3) Quel est l'élément unitaire de l'image ? Quelle est sa dimension ?
4) La valeur radiométrique du pixel est comprise entre quelles valeurs ?
5) Quelle est la taille d'une image ?

Exo. 2 : Déclaration et initialisation de variables

1) Fonction des trois fenêtres
2) Choix du répertoire de travail
3) Commandes dir, cd .., del
4) scalaire y=1
5) vectorielle y=[1 3 2], y=[1 ;3 ;2]
6) matricielle y=[1 3 6 ;8 4 7]
7) y(2,2)= ?
8) fonction du point virgule en fin de ligne ?
9) initialisation à 0 y=zeros(3,1) y=zeros(3) y=zeros(3,4)
10) initialisation à 1 y=ones(3,1) y=ones(3) y=ones(3,4)
11) Création d'une matrice 10lignes, 10 colonnes, 3 bandes dont le pixel (5,5) vaut 10 dans les trois bandes

Exo. 3 : Opération entre variables

1) initialisation x=[1 2 3] y=[3 4 5]
2) transposée z=y'
3) calcul de z : z=x+y z=x-y z=x*y z=x*y' z=x.*y
4) z= x+1 z=x*5

Exo. 4 : Lecture affichage et écriture d'une image

1) Lecture a=imread('cameraman.tif')
2) Quelle est la dimension de a
3) Lecture b=imread('autumn.tif')
4) Quelle est la dimension de b
5) Affichage imshow(a)
6) subplot(1,2,1)
   imshow(a)
   subplot(1,2,2)
   imshow(b)
7) Modifier b pour obtenir un carré blanc au milieu de l'image de taille 50*30 pixels
8) Ecriture Imwrite(b,'nom.tif')

Commandes Matlab
en traitement des images

Affichage

figure(#)	Ouvre une nouvelle figure
imshow	Affiche une image
pixval	Affiche sur la figure les valeurs de ligne, colonne et la valeur des pixels dans les 3 plans
impixel	Entre dans une variable les valeurs des pixels que l'on a cliqués dans l'image
improfile	Entre dans une variable les valeurs des pixels d'un segment
subplot	Permet d'afficher dans une figure plusieurs images ou graphes
title	Donne un titre à une image ou à un graphe

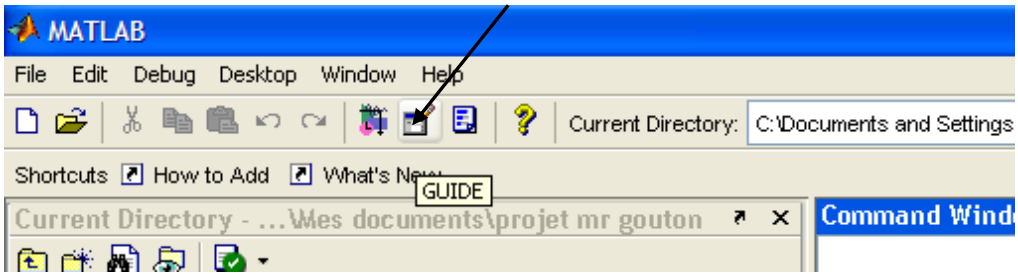
Fonctions

rgb2grey	Convertit une image couleur en niveaux de gris
filter	Applique un filtre linéaire une image
fft2	Calcule la transformée de fourier d'une image (résultat en complexe)
fftshift	Recentre les basses fréquences au centre de l'image
ifft2	Calcule la transformée de fourier inverse d'une image (résultat en complexe)
imhist	Affiche l'histogramme d'une image en niveaux de gris
imnoise	Ajoute du bruit à une image
corr2	Calcule la corrélation entre deux images
min	Calcule la valeur min d'un vecteur
max	Calcule la valeur max d'un vecteur
zeros	Initialise une variable à zero
Ones	Initialise une variable à un
Size	Donne la taille d'une image (ligne, colonne, bandes)

Création de l'interface graphique (Guide en matlab) :

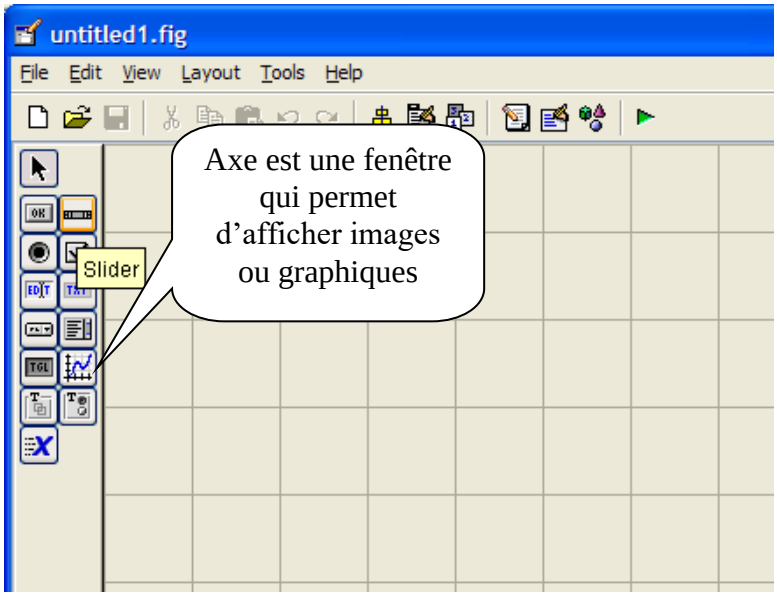
- lancer *MATLAB*

- cliquer sur l'icône « GUIDE » (voir la figure ci-après)



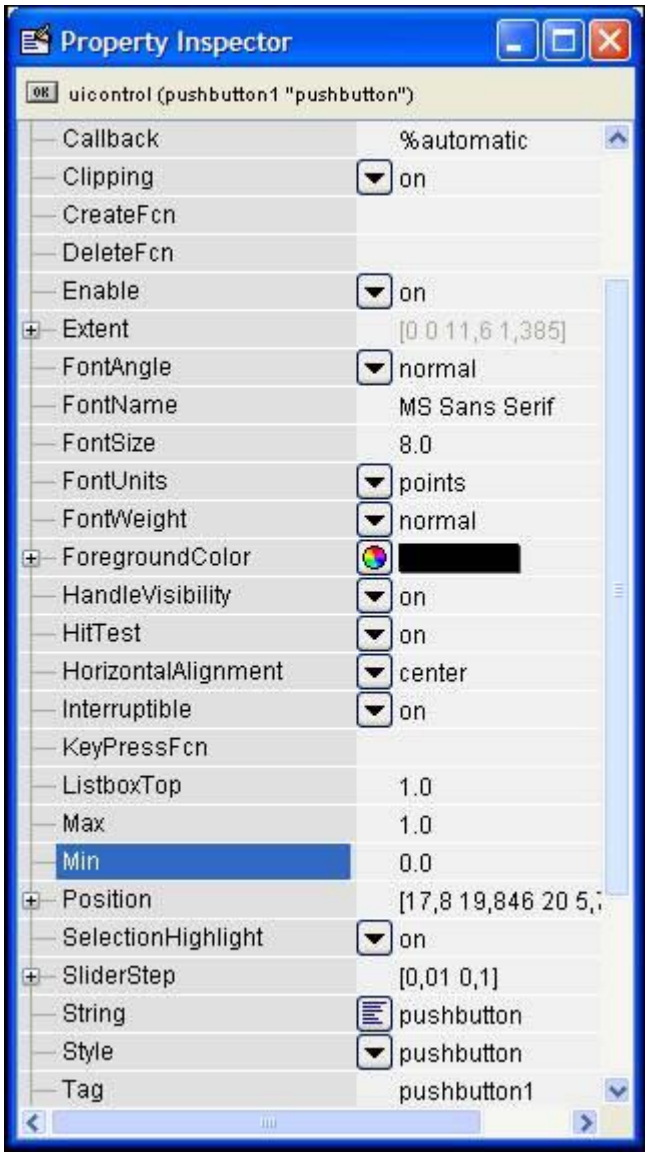
- dans l'onglet « Create new GUI » choisir un des quatre modes (par défaut on prendra « Blank GUI »

- on obtient une fenêtre avec sur la gauche les différents objets (push button, radio button, slider etc…) que l'on peut placer sur l'interface graphique



- Il suffit de choisir un objet et de le dessiner sur l'interface graphique

- Pour accéder aux paramètres de cet objet, on double clique sur cet objet et une fenêtre *Property Inspector* s'ouvre :



Les paramètres à modifier sont :

- String : le nom qui apparaît sur le bouton
- Tag : cela modifie le nom de la fonction dans *MATLAB* associé au bouton

Important : bien s'assurer que le nom dans « Callback » est bien modifié et est en rapport avec le « Tag ».

- Dès que l'on sauvegarde notre fichier « GUIDE », *MATLAB* génère automatiquement un fichier .m portant le même nom que le fichier « GUIDE ». La partie surlignée est la partie créée à partir du bouton. Il suffit de rajouter du code à la suite de cette partie pour ajouter des actions associées au bouton.

1. Ouverture d'une image

Créer un bouton ouverture d'une image

Dans la fonction correspondant au tag écrire

```
name_file=uigetfile( '*.img')
Fid = fopen(name_file);
L = inputdlg('Lignes');
C = inputdlg('Colonnes');
B = inputdlg('Bandes');
L = str2double(L);
B = str2double(B);
C = str2double(C);
donnees = fread(Fid, C * L * B,'*ubit8');
fclose(Fid);
donnees = reshape(donnees, C, L, B);
donnees(:,:,1)=donnees(:,:,1)';
donnees(:,:,2)=donnees(:,:,2)';
donnees(:,:,3)=donnees(:,:,3)';


IM(:,:,1)=donnees(:,:,3);
IM(:,:,2)=donnees(:,:,2);
IM(:,:,3)=donnees(:,:,1);
```

Dans la même fonction afficher l'image en couleur avec la fonction imshow.

Ecrire une fonction histogramme qui retourne l'histogramme d'une image en niveau de gris et afficher dans une seconde fenêtre l'histogramme des 3 canaux (rouge, vert et bleu). Voir page de garde du fascicule.

Commenter.

Avec la fonction assignin, créer une variable globale qui contient l'image et qui pourra être récupérée par les autres fonctions.

```
assignin('base', 'image',IM);
```

2. Transformations :

Créer un panel « Transformations »

a. Transformation linéaire

-Récupérer l'image avec la commande « evalin »
Comme l'image est sombre, ajouter un bouton « Transformation linéaire » créer une fonction (à part) qui retourne une image en niveaux de gris dont la valeur min est à 0 et la valeur max à 255. L'appliquer à chaque bande et afficher le résultat en couleur et ses 3 histogrammes.
- Commenter.

Vous utiliserez cette image transformée linéairement en entrée de tous les autres traitements du TP

```
assignin('base', 'image',IM);
```

b. Egalisation

-Ajouter dans transformation, un bouton Egalisation
- Ecrire (à part) une fonction Egalisation.m qui réalise l'égalisation d'une image en niveau de gris et qui retourne une image égalisée
- L'appliquer à chaque bande et afficher l'image égalisée en couleur ainsi que son histogramme (1 par bande)
- Commenter

3. Contours :

a. Sous forme de menu, proposer un filtrage de Prewitt, horizontal, vertical et mixte

L'appliquer à l'image originale et commenter.
Comment peut-on améliorer le résultat ?

b. Faire de même pour le filtrage de Sobel, horizontal, vertical et mixte

c. Comparer les 2 détecteurs de contours

4. Filtrage de bruit :

a. Proposer différents types de bruit (gaussien, poisson, poivre et sel)
b. Proposer différents types de filtrages (passe bas, médian, milieu, Fourier)
c. Comparer les performances des filtres sur les différents types de bruits

5. Seuillages :

a. Binarisation

- On souhaite réaliser un masque sur l'eau
- Demander à l'utilisateur de donner un seuil avec un curseur entre 0 et 255 que l'on appliquera à la bande 1 (PIR).
- Ecrire une fonction binarisation.m
- Afficher le résultat et son histogramme

b. Seuillage multiseuils en niveau de gris

- A partir de l'image en niveau de gris
- Demander à l'utilisateur le nombre de niveaux requis en sortie (entre 0 et 255)

- Ecrire une fonction seuillage qui calcule automatiquement les seuils équi-répartis entre 0 et 255 pour définir les n niveaux

- Afficher le résultat et son histogramme, vérifier que sur l'histogramme, on a bien le nombre de niveaux attendus.

c. Seuillage multiseuils en couleur

- Demander à l'utilisateur le nombre de niveaux requis

- Ecrire une fonction seuillage qui calcule automatiquement les seuils équi-répartis entre 0 et 255 pour définir les n^3 niveaux

- Vérifier que les couleurs sont cohérentes

- Afficher les histogrammes des 3 plans de l'image d'entrée et les histogrammes des 3 plans de l'image de sortie.

6. Morphologie mathématique :

- Créer des boutons qui effectuent une érosion, une dilation, une ouverture et une fermeture de l'image d'entrée en niveaux de gris

a. Application au filtrage de bruit

- Ajouter un bruit 'poivre et sel' à l'image d'entrée

- Appliquer une érosion ou une dilatation et observer le résultat

- Imaginer une combinaison pour filtrer totalement le bruit

- Créer un bouton à cet effet dans l'IHM (filtrage)

b. Application à la détection de contours

- A partir de l'image en niveau de gris, appliquer une combinaison de commandes pour mettre en évidence les contours.

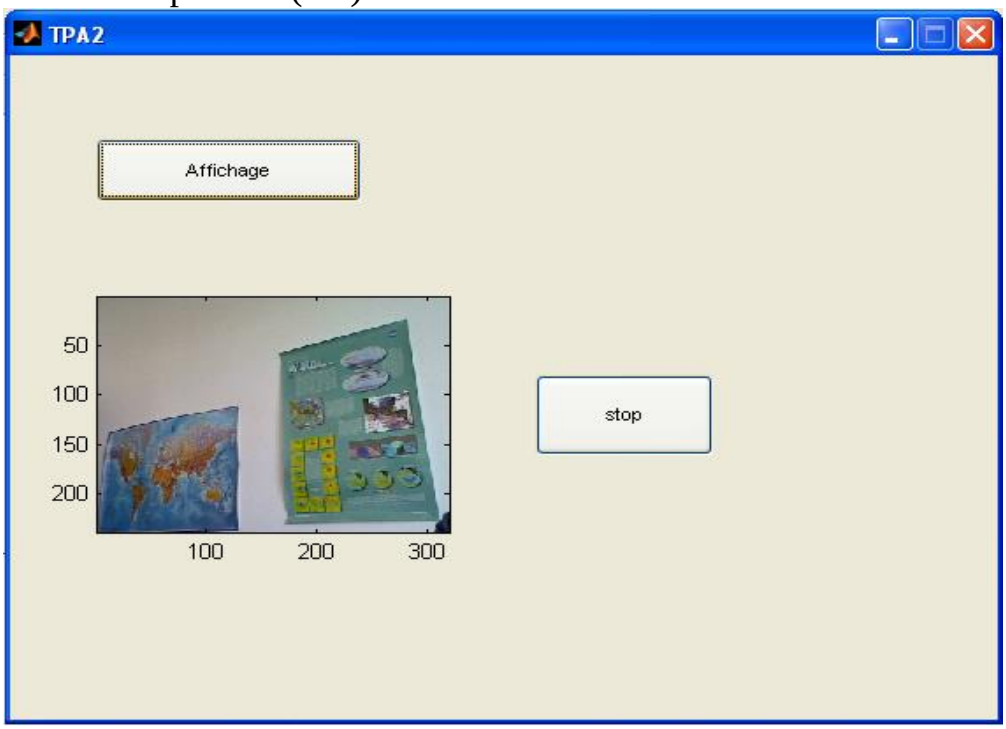
- Créer un bouton à cet effet dans l'IHM (contours)

7. Tester votre IHM sur d'autres images couleur

8. Adapter votre IHM à la connexion à une webcam

Pour connaitre les paramètres de la caméra taper successivement les commandes suivantes :

- imaqhwinfo
- info=imaqhwinfo('winvideo')
- info.DeviceInfo.SupportedFormats
- vid = videoinput('winvideo',3,'RGB24_640x480');
- preview(vid)




```
function Affichage_Callback(hObject, eventdata, handles)
%guidata(hObject, handles);

set(handles.Stop,'UserData',0);
vid = videoinput('winvideo',1,'YUY2_320x240'); % à adapter à VOTRE caméra
set(vid,'FramesPerTrigger',1);
set(vid,'TriggerRepeat',Inf);
triggerconfig(vid, 'Manual');
start(vid);
trigger(vid);

y = (getdata(vid,1,'uint8'));

while ~(get(handles.Stop,'UserData'))
    trigger(vid)
    y = (getdata(vid,1,'uint8'));
```

```
    axes(handles.axes1)
    subimage(y)
end
stop(vid);
return

------------------------------------------------------------------
function Stop_Callback(hObject, eventdata, handles)
set(handles.Stop,'enable','on');
set(handles.Stop,'UserData',1);
%guidata(hObject, handles);
```